

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Loading data from the CSV file

```
In [ ]: df = pd.read_csv("data.csv")
df
```

```
Out[ ]:
```

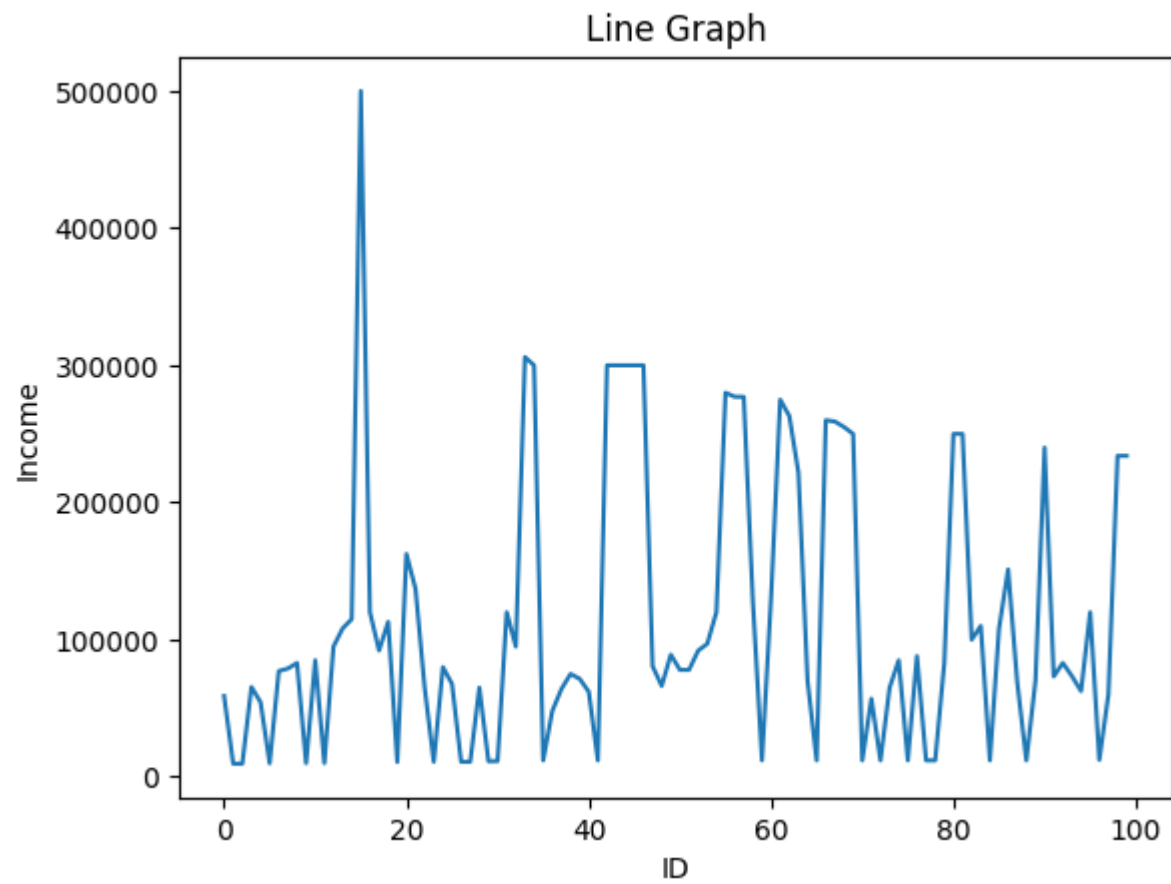
	<b>Id</b>	<b>Age</b>	<b>Income</b>	<b>Home</b>	<b>Emp_length</b>	<b>Intent</b>	<b>Amount</b>	<b>Rate</b>	<b>Status</b>	<b>Percent_income</b>	<b>Default</b>	<b>Cred_length</b>
<b>0</b>	0	22	59000	RENT	123	PERSONAL	35000	16.02	1	0.59	Y	3
<b>1</b>	1	21	9600	OWN	5	EDUCATION	1000	11.14	0	0.10	N	2
<b>2</b>	2	25	9600	MORTGAGE	1	MEDICAL	5500	12.87	1	0.57	N	3
<b>3</b>	3	23	65500	RENT	4	MEDICAL	35000	15.23	1	0.53	N	2
<b>4</b>	4	24	54400	RENT	8	MEDICAL	35000	14.27	1	0.55	Y	4
<b>...</b>	...	...	...	...	...	...	...	...	...	...	...	...
<b>95</b>	95	23	120000	RENT	1	EDUCATION	25600	12.69	0	0.21	N	3
<b>96</b>	96	24	12360	OWN	2	MEDICAL	1600	13.57	0	0.13	N	3
<b>97</b>	97	22	60000	RENT	0	VENTURE	25475	10.99	1	0.42	N	3
<b>98</b>	98	25	234000	MORTGAGE	3	MEDICAL	20000	14.27	0	0.09	Y	4
<b>99</b>	99	24	234000	OWN	8	HOMEIMPROVEMENT	20000	8.88	0	0.09	N	4

100 rows × 12 columns

Line Graph

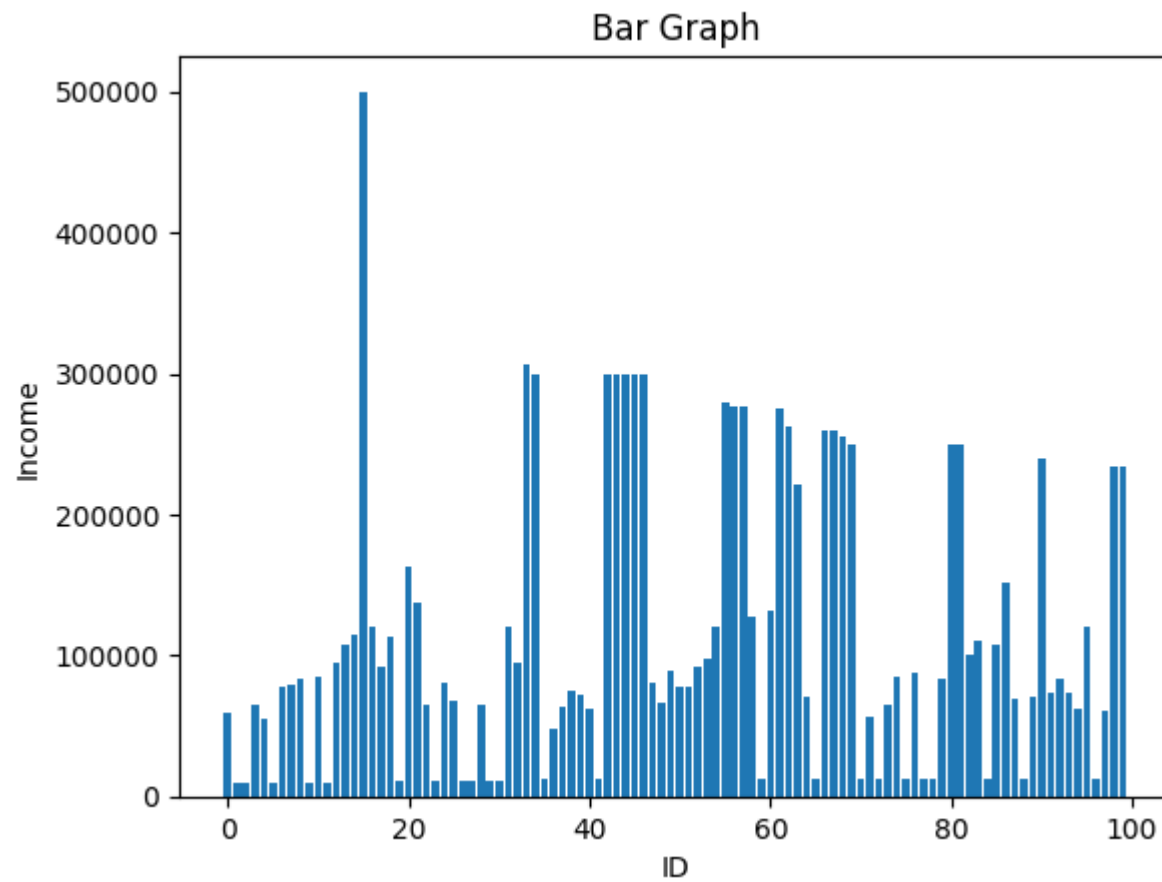
```
In [ ]: plt.plot(df['Id'], df['Income'])
plt.title('Line Graph')
```

```
plt.xlabel('ID')  
plt.ylabel('Income')  
plt.show()
```



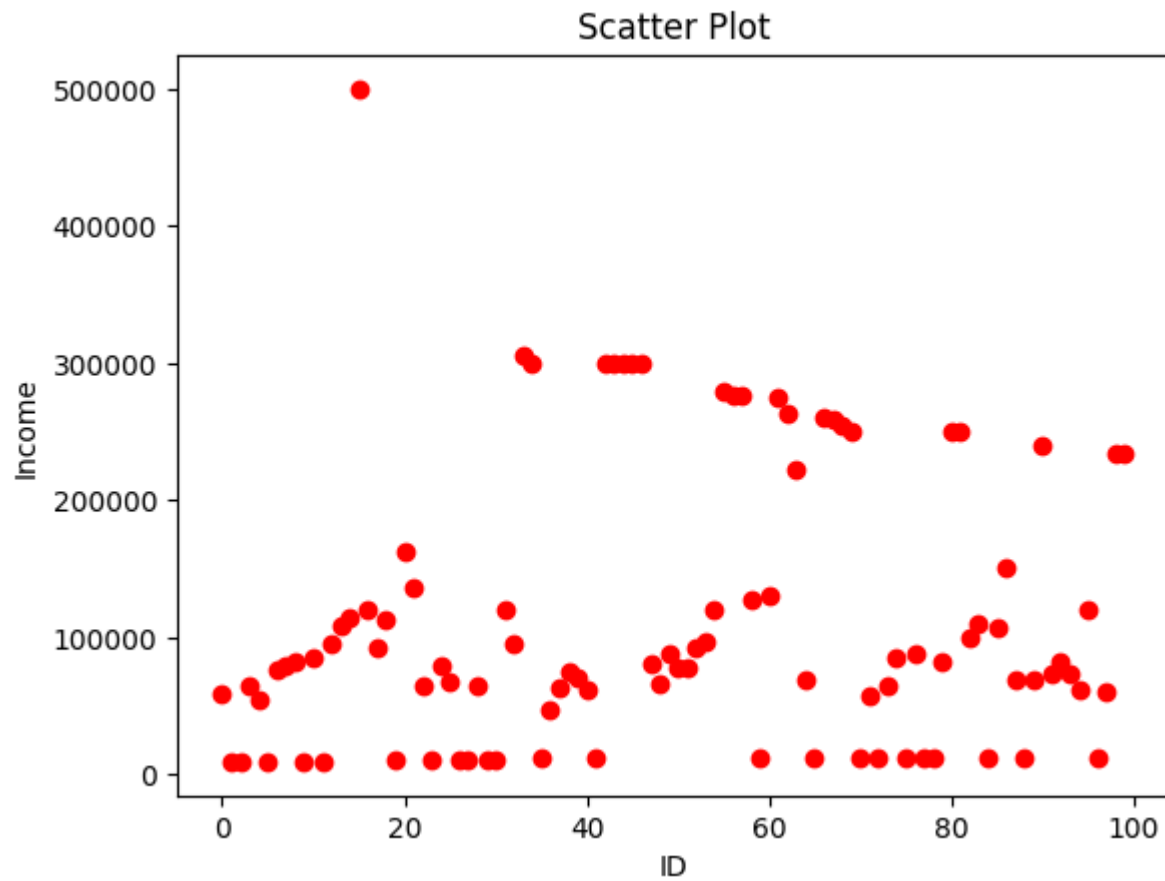
Bar Graph

```
In [ ]: plt.bar(df['Id'], df['Income'])  
plt.title('Bar Graph')  
plt.xlabel('ID')  
plt.ylabel('Income')  
plt.show()
```



Scatter Plot

```
In [ ]: plt.scatter(df['ID'], df['Income'], color='r')
plt.title('Scatter Plot')
plt.xlabel('ID')
plt.ylabel('Income')
plt.show()
```



Correlation Heatmap

```
In [ ]: df_encoded = pd.get_dummies(df, columns=["Home", "Intent", "Status", "Default"], drop_first=True)

# Calculate the correlation matrix for all numeric columns (including the encoded ones)
correlation_matrix = df_encoded.corr()

# Plot the heatmap using matshow with a colormap
cax = plt.matshow(correlation_matrix, cmap='coolwarm')

# Add a colorbar to the heatmap
plt.colorbar(cax)
```

```
# Set X-axis and Y-axis Labels using column names
plt.xticks(range(len(correlation_matrix.columns)), correlation_matrix.columns, rotation=90)
plt.yticks(range(len(correlation_matrix.columns)), correlation_matrix.columns)

# Set the title of the heatmap
plt.title('Correlation Heatmap')

# Show the heatmap
plt.show()
```

### Correlation Heatmap

